Quantum Algorithms for the Jones Polynomial

Louis H. Kauffman[a] and Samuel J. Lomonaco Jr.[b]

[a] Department of Mathematics, Statistics and Computer Science (m/c 249), 851 South Morgan Street, University of Illinois at Chicago, Chicago, Illinois 60607-7045, USA

[b] Department of Computer Science and Electrical Engineering, University of Maryland Baltimore County, 1000 Hilltop Circle, Baltimore, MD 21250, USA

ABSTRACT

This paper gives a generalization of the AJL algorithm for quantum computation of the Jones polynomial to continuous ranges of values on the unit circle for the Jones parameter. We show that the Kauffman-Lomonaco 3-strand algorithm for the Jones polynomial is a special case of this generalization of the AJL algorithm.

Keywords: knots, links, braids, quantum computing, unitary transformation, Jones polynomial, Temperley-Lieb algebra

1. INTRODUCTION

In[11] and in[14] we gave a quantum algorithm for computing the Jones polynomial via a unitary representation of the three-strand Artin braid group to the Temperley-Lieb algebra. In the bracket polynomial version of this representation (see Section 2 of the present paper) the representations were unitary for certain continuous ranges of choice of the polynomial variable A on the unit circle in the complex plane. In this paper we show that these three-strand representations are a subset of unitary representations of the Artin braid group on arbitrary numbers of strands and corresponding continuous ranges of the variable A on the unit circle. These more general representations are in fact generalizations of the AJL representations[1,2] that were originally defined at certain roots of unity in the unit circle.

The paper is organized as follows. In Section 2 we review the bracket polynomial model for the Jones polynomial, and its relationship with representations of the Temperley-Leib algebra. In Section 3 we review the 3-strand representation. In Section 4 we detail diagrammatically the construction of the generalized representation and show how it is related to the 3-strand representation and to the AJL representation. In Section 5 we give a diagrammatic proof of the requisite trace formula that is needed to make this representation into a quantum algorithm for computing the Jones polynomial. Much remains to be explored in these directions. The present paper was sparked by our work in[16] on NMR quantum computing, and there will be a sequel to the present paper[17] that relates the present work to NMR research.

2. BRACKET AND TEMPERLEY LIEB ALGEBRA

The bracket polynomial[7] model for the Jones polynomial[4–6,19] is usually described by the expansion

$$\langle \times \rangle = A\langle \,)(\, \rangle + A^{-1}\langle \asymp \rangle \tag{1}$$

and we have

$$\langle K \bigcirc \rangle = (-A^2 - A^{-2})\langle K \rangle \tag{2}$$

$$\langle \gamma \rangle = (-A^3)\langle \smile \rangle \tag{3}$$

$$\langle \gamma \rangle = (-A^{-3})\langle \smile \rangle \tag{4}$$

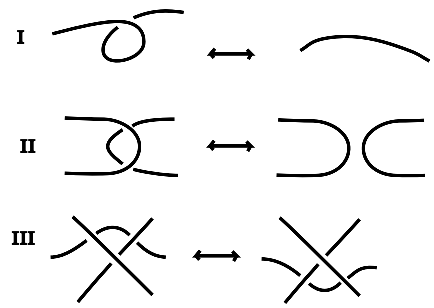


Figure 1. Reidemeister Moves

The bracket expansion of a knot or link diagram is invariant under Reidemeister moves II and III as shown in Figure 1, and can be normalized to be invariant under the first Reidemeister moves by multiplication by an appropriate power of $-A^3$. Once normalized, it is a version of the Jones polynomial,[4] differing from it by a simple change of variable.

The key idea behind the present quantum algorithms to compute the Jones polynomial is to use unitary representations of the braid group derived from Temperley-Lieb algebra representations that take the form

$$\rho(\sigma_i) = AI + A^{-1}U_i$$

where σ_i is a standard generator of the Artin braid group, A is a complex number of unit length, and U_i is a symmetric real matrix that is part of a representation of the Temperley-Lieb algebra. A diagrammatic version of the Temperley-Lieb algebra puts the form of this representation in exact correspondence with the bracket expansion, where the parallel arcs)(correspond to the identity element of the algebra and the arcs in the form correspond to the generator U_i of the algebra when corresponds to the braid generator σ_i. For more details about this strategy and the background information about the Jones polynomial, the bracket model for the Jones polynomial and the Temperley-Lieb algebra the reader may wish to consult.[1, 2, 4, 7–15, 18] In the following sections, we have made use of such diagrammatic techniques and have included some material to make the paper partly self-contained.

3. TWO PROJECTORS AND A UNITARY REPRESENTATION OF THE THREE STRAND BRAID GROUP

It is useful to think of the Temperley Lieb algebra as generated by projections $e_i = U_i/\delta$ so that $e_i^2 = e_i$ and $e_i e_{i\pm 1} e_i = \tau e_i$ where $\tau = \delta^{-2}$ and e_i and e_j commute for $|i-j| > 1$.

With this in mind, consider elementary projectors $e = |A\rangle\langle A|$ and $f = |B\rangle\langle B|$. We assume that $\langle A|A\rangle = \langle B|B\rangle = 1$ so that $e^2 = e$ and $f^2 = f$. Now note that

$$efe = |A\rangle\langle A|B\rangle\langle B|A\rangle\langle A| = \langle A|B\rangle\langle B|A\rangle e = \tau e$$

Thus

$$efe = \tau e$$

where $\tau = \langle A|B\rangle\langle B|A\rangle$.

Further author information: L.H.K. E-mail: kauffman@uic.edu, S.J.L. Jr.: E-mail: lomonaco@umbc.edu

This algebra of two projectors is the simplest instance of a representation of the Temperley Lieb algebra. In particular, this means that a representation of the three-strand braid group is naturally associated with the algebra of two projectors.

Quite specifically if we let $\langle A| = (a, b)$ and $|A\rangle = (a, b)^T$ the transpose of this row vector, then

$$e = |A\rangle\langle A| = \begin{bmatrix} a^2 & ab \\ ab & b^2 \end{bmatrix}$$

is a standard projector matrix when $a^2 + b^2 = 1$. To obtain a specific representation, let

$\mathrm{l}e_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$ and $e_2 = \begin{bmatrix} a^2 & ab \\ ab & b^2 \end{bmatrix}$.

It is easy to check that $e_1 e_2 e_1 = a^2 e_1$ and that $e_2 e_1 e_2 = a^2 e_2$.

Note also that $e_1 e_2 = \begin{bmatrix} a^2 & ab \\ 0 & 0 \end{bmatrix}$ and $e_2 e_1 = \begin{bmatrix} a^2 & 0 \\ ab & 0 \end{bmatrix}$.

We define $U_i = \delta e_i$ for $i = 1, 2$ with $a^2 = \delta^{-2}$. Then we have , for $i = 1, 2$

$$U_i^2 = \delta U_i \, , \; U_1 U_2 U_1 = U_1, \; U_2 U_1 U_2 = U_2.$$

Thus we have a representation of the Temperley-Lieb algebra on three strands. See[9] for a discussion of the properties of the Temperley-Lieb algebra.

Note also that we have

$$trace(U_1) = trace(U_2) = \delta,$$

while

$$trace(U_1 U_2) = trace(U_2 U_1) = 1$$

where *trace* denotes the usual matrix trace. These formulas show that the trace is working correctly with respect to the bracket evaluation. See the last part of the present paper for a more extended discussion of this point.

Now we return to the matrix parameters: Since $a^2 + b^2 = 1$ this means that $\delta^{-2} + b^2 = 1$ whence $b^2 = 1 - \delta^{-2}$. Therefore b is real when δ^2 is greater than or equal to 1.

We are interested in the case where $\delta = -A^2 - A^{-2}$ and A *is a unit complex number*. Under these circumstances the braid group representation

$$\rho(\sigma_i) = AI + A^{-1} U_i$$

will be unitary whenever U_i is a real symmetric matrix. Thus we will obtain a unitary representation of the three-strand braid group B_3 when $\delta^2 \geq 1$.

For any A with $d = -A^2 - A^{-2}$ these formulas define a representation of the braid group. With $A = exp(i\theta)$, we have $d = -2cos(2\theta)$. We find a specific range of angles θ in the following disjoint union of angular intervals

$$\theta \in [0, \pi/6] \sqcup [\pi/3, 2\pi/3] \sqcup [5\pi/6, 7\pi/6] \sqcup [4\pi/3, 5\pi/3] \sqcup [11\pi/6, 2\pi]$$

that give unitary representations of the three-strand braid group. Thus a specialization of a more general represention of the braid group gives rise to a continuous family of unitary representations of the braid group.

3.1. A Quantum Algorithm for the Jones Polynomial on Three Strand Braids

We gave above an example of a unitary representation of the three-strand braid group.In fact, we can use this representation to compute the Jones polynomial for closures of 3-braids, and therefore this representation provides a test case for the corresponding quantum computation. We now analyse this case by first making explicit how the bracket polynomial is computed from this representation. This unitary representation and its application to a quantum algorithm first appear in.[11] When coupled with the Hadamard test, this algorithm gets values for the Jones polynomial in polynomial time in the same way as the AJL algorithm.[1] It remains to be seen how fast these algorithms are in principle when asked to compute the polynomial itself rather than certain specializations of it.

First recall that the representation depends on two matrices U_1 and U_2 with

$$U_1 = \begin{bmatrix} \delta & 0 \\ 0 & 0 \end{bmatrix} \text{ and } U_2 = \begin{bmatrix} \delta^{-1} & \sqrt{1-\delta^{-2}} \\ \sqrt{1-\delta^{-2}} & \delta - \delta^{-1} \end{bmatrix}.$$

The representation is given on the two braid generators by

$$\rho(\sigma_1) = AI + A^{-1}U_1$$
$$\rho(\sigma_2) = AI + A^{-1}U_2$$

for any A with $\delta = -A^2 - A^{-2}$, and with $A = exp(i\theta)$, then $\delta = -2cos(2\theta)$. We get the specific range of angles $\theta \in [0, \pi/6] \sqcup [\pi/3, 2\pi/3] \sqcup [5\pi/6, 7\pi/6] \sqcup [4\pi/3, 5\pi/3] \sqcup [11\pi/6, 2\pi]$ that give unitary representations of the three-strand braid group.

Note that $tr(U_1) = tr(U_2) = \delta$ while $tr(U_1U_2) = tr(U_2U_1) = 1$. If b is any braid, let $I(b)$ denote the sum of the exponents in the braid word that expresses b. For b a three-strand braid, it follows that

$$\rho(b) = A^{I(b)}I + \tau(b)$$

where I is the 2×2 identity matrix and $\tau(b)$ is a sum of products in the Temperley Lieb algebra involving U_1 and U_2. Since the Temperley Lieb algebra in this dimension is generated by I,U_1, U_2, U_1U_2 and U_2U_1, it follows that

$$\langle \overline{b} \rangle = A^{I(b)}\delta^2 + tr(\tau(b))$$

where $\overline{b}$ denotes the standard braid closure of b, and the sharp brackets denote the bracket polynomial as described in previous sections. From this we see at once that

$$\langle \overline{b} \rangle = tr(\rho(b)) + A^{I(b)}(\delta^2 - 2).$$

It follows from this calculation that the question of computing the bracket polynomial for the closure of the three-strand braid b is mathematically equivalent to the problem of computing the trace of the matrix $\rho(b)$.

The matrix in question is a product of unitary matrices, the quantum gates that we have associated with the braids σ_1 and σ_2. The entries of the matrix $\rho(b)$ are the results of preparation and detection for the two dimensional basis of qubits for our machine:

$$\langle i|\rho(b)|j\rangle.$$

Given that the computer is prepared in $|j\rangle$, the probability of observing it in state $|i\rangle$ is equal to $|\langle i|\rho(b)|j\rangle|^2$. Thus we can, by running the quantum computation repeatedly, estimate the absolute squares of the entries of the matrix $\rho(b)$. This will not yield the complex phase information that is needed for either the trace of the matrix or the absolute value of that trace.

However, we do know how to write a quantum algorithm to compute the trace of a unitary matrix (via the Hadamard test). Since $\rho(b)$ is unitary, we can use this approach to approximate the trace of $\rho(b)$. This yields a quantum algorithim for the Jones polynomial for three-stand braids (evaluated at points A such that the representation is unitary). Knowing $tr(\rho(b))$ from the quantum computation, we then have the formula for the bracket, as above,

$$\langle\overline{b}\rangle = trace(\rho(b)) + A^{I(b)}(\delta^2 - 2).$$

Then the normalized polynomial, invariant under all three Reidemeister moves is given by

$$f(\overline{b}) = (-A^3)^{-I(b)}\langle\overline{b}\rangle.$$

Finally the Jones polynomial in its usual form is given by the formula

$$V(\overline{b})(t) = f(\overline{b})(t^{-1/4}).$$

Thus we conclude that our quantum computer can approximate values of the Jones polynomial.

4. GENERALIZING THE AJL REPRESENTATION

In this section we show how the KL (Kauffman-Lomonaco) algorithm described in the previous section becomes a special case of a generalization of the AJL algorithm: Here we use notation from the AJL paper. In that paper, the generators U_i (in our previous notation) for the Temperley-Lieb algebra, are denoted by E_i. We will first describe the AJL representation of the Temperley-Lieb Algebra and we will show how that representation works for a continuous range of values of the parameter θ described below. In the original treatment of AJK[1] the values are restricted to a discrete range corresponding to $exp(i\theta/2)$ a root of unity. We observed this phenomena of continuous ranges of values in our work on the three-strand model, described in the previous section, and found a way to extend it to AJL, as will be detailed below. This work has benefited from interaction with all of the authors of the paper[16] and our collaboration in bringing these algorithms to application in NMR quantum computing. The contents of this section will be connected with NMR quantum computing in a paper that is under preparation.[17]

In this section we will construct matrix representations of the Temperley-Lieb algebra. In the next section, we will discuss the structure of trace functions on these representations that can be used to produce quantum algorithms.

Let $\lambda_k = sin(k\theta)$. For the time being θ is an arbitrary angle. Let $A = iexp(i\theta/2)$ so that $d = -A^2 - A^{-2} = 2cos(\theta)$.

We need to choose θ so that $sin(k\theta)$ is non-negative for the range of k's we use (these depend on the choice of line graph as in AJL). And we insist that $sin(k\theta)$ is non-zero except for $k = 0$. Then it follows from trigonometry that for all k

$$d = (\lambda_{k-1} + \lambda_{k+1})/\lambda_k.$$

We shall see, below, that the values of k will range from 0 to r for a fixed $r > 2$ in a given representation. We ask that $sin(r\theta)$ be greater than zero and can take θ in the continuous range $0 < \theta \leq \pi/r$.

The AJL representation of the Temperley-Lieb algebra is based on the complex vector space H_n whose basis is $\{|i\rangle\}$ where i is a (0, 1) bitstring of length n. Each bitstring i is seen as corresponding to a walk on a line-graph G_r with $r - 2$ edges and $r - 1$ nodes. For example, the graph below is G_5.

$$1 - - - - - 2 - - - - - 3 - - - - - 4$$

Bitstrings represent walks on a line graph with 0 corresponding to a step to the left and 1 corresponding to a step to the right. The walk begins at the left-most node, labelled 1. Thus 1011 represents the walk Right, Left, Right, Right ending at node number 3 in the line graph above. We say that a string of n bits is a *walk on* G_r if the walk remains inside G_r for all its steps (i.e. one never faces the instruction to go beyond the right-most or left-most end points of the graph). We let $H_{n,r}$ denote the subspace of H_n spanned by bitstrings that are walks on G_r.

The representation of the Temperley-Lieb algebra is denoted by

$$\Phi : TL_n \longrightarrow Matr(H_{n,r})$$

where TL_n denotes the n-strand Temperley-Lie algebra generated by $I, E_1, \cdots, E_{n-1}$ and $Matr(H_{n,r})$ denotes matrix mappings of this space in the given bitstring basis. The AJL representation is given in terms of E_i such that $E_i^2 = dE_i$ and the E_i satisfy the Temperley-Lieb relations. Each E_i acts non-trivially at the i and $i+1$ places in the bit-string basis for the space. If p denotes the bitstring and we are computing $\Phi(|p\rangle)$, then we define $z(i)$ to be the endpoint of the walk described by the bitstring p using only the first $(i-1)$ bits of p. Each $E_i(p)$ is based upon $\lambda_{z(i)-1}, \lambda_{z(i)}, \lambda_{z(i)+1}$. In our example we have $p = 1011$ represents the walk Right, Left, Right, Right ending at node number 3 in

$$1 - - - - - 2 - - - - - 3 - - - - - 4,$$

and $z(1) = 1, z(2) = 2, z(3) = 1, z(4) = 1, z(5) = 3$.

More precisely, if we let

$$|v(a)\rangle = (\sqrt{\lambda_{a-1}/\lambda_a}, \sqrt{\lambda_{a+1}/\lambda_a})^T$$

(i.e. this is a column vector. T denotes transpose.) Then

$$E_i = |v(z(i))\rangle\langle v(z(i))|.$$

Here it is understood that this refers to the action on the bitstrings

$$- - - - - - - - - - - 01 - - - - - - - - - - -$$

and

$$- - - - - - - - - - - 10 - - - - - - - - - - -$$

obtained from the given bitstring by modifying the i and $i+1$ places. The basis order is 01 before 10.

The explicit form of the the transformations E_i is given by the equations below for a generic matrix E. We need to explicate $\Phi(E_i)$ and we shall simply write E_i instead of $\Phi(E_i)$.

$$E = vv^T = \begin{bmatrix} \lambda_-/\lambda_0 & \frac{\sqrt{\lambda_-\lambda_+}}{\lambda_0} \\ \frac{\sqrt{\lambda_-\lambda_+}}{\lambda_0} & \lambda_+/\lambda_0 \end{bmatrix}$$

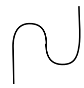
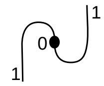
Here $v^T = (\sqrt{\frac{\lambda_-}{\lambda_0}}, \sqrt{\frac{\lambda_+}{\lambda_0}})$. It is easy to see that

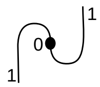
$$E^2 = (\frac{\lambda_- + \lambda_+}{\lambda_0})E$$

since $E^2 = vv^Tvv^T = (v^Tv)E$. For E_i, we take $\lambda_- = \lambda_{z(i)-1}$, $\lambda_+ = \lambda_{z(i)+1}$ and $\lambda_0 = \lambda_{z(i)}$. For the action of $E_i(p)$ for a given bitstring p, let $p|i$ denote the restriction of p to the first $i-1$ bits in the string. Then we need to explicate E_i at the bitstring p and hence its values at $p|i\,k\,l\rangle$ where k and l are either 0 or 1. The transformation will not change bits beyond these two extra places. Then *by definition*

$$E_i|p|i\,0\,0\rangle = E_i|p|i\,1\,1\rangle = 0$$

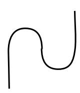
and E_i acts on the remaining subspace according to the matrix formulas we have given above. This means that we can regard E_i diagrammatically as a cup-cap combination. $E = \begin{matrix}\cup \\ \cap\end{matrix}$ where the legs of the diagram correspond to the indices that can be either 0 or 1. In this formalism $\cap$ takes the role of v^T and $\cup$ the role of v in the decomposition $E = vv^T$. Each cap or cup can receive only two indices and these must be different,

since the transformations are zero when there is a repetition of 0 or a repetition of 1. The only further relation that is needed to prove from this diagrammatic point of view is that  is an identity transformation as a mapping defined on a single bit. The relations $E_iE_{i+1}E_i = E_i$ follow from this. To see that this composition is the identity, consider one of its cases as shown here: 

. The fact that each cup and each cap can only support a zero and a one, shows that the composition will necessarily be a multiple of the identity. To see the details, we note that the local binary bit sequence must match the indices, which in this case is 101. This means that $z(i+1) = z(i) + 1$ (due to the starting 1 in the local bitstring, shifting to the next node in the graph). The scalar contribution of 

is equal to

$$S = \sqrt{\frac{\lambda_{z(i)+1}}{\lambda_{z(i)}} \frac{\lambda_{z(i+1)-1}}{\lambda_{z(i+1)}}}$$

where the first factor under the square root comes from the cap, and the second factor under the square root comes from the cup. The zero in the string 101 corresponds to the $z(i+1) - 1$ in the second factor. Since we know that $z(i+1) = z(i) + 1$ it follows that $S = 1$. This is one of the small number of cases to check that proves that  is the identity transformation. This completes the proof that $\Phi : TL_n \longrightarrow Matr(H_{n,r})$ is a representation of the Temperley-Lieb algebra.

Remark. As we have seen in the discussion above, it helps to take a diagrammatic point of view. Here we remark via Figure 2 on a simpler representation of the Temperley-Lieb algebra that is analogous to the AJL representation that we have discussed here. In Figure 2 we illustrate diagrams for the basic elements of the Temperley-Lieb algebra as we have explained them in the previous section. In order to build the elements $U_i = E_i$ we make them diagrammatically – each a combination of a cup and a cap (and appropriate identity lines), as illustrated in this figure. In the representation illustrated in this figure, each cup and each cap is represented by the same matrix M and the necessary conditions for that M are that the sum of the squares of its entries should be equal to $d = 2cos(\theta)$ (as above), and that the matrix product of the cup and cap matrices should be equal to the identity matrix. This is illustrated in the middle of Figure 2 by the identity showing the wavy line being pulled straight to form an identity line. All the diagrams in the figure correspond to matrices, with the indices for matrix elements corresponding to labelled endpoints of the lines in the diagram. When two diagrammatic matrices are composed, an output line of one matrix is attached to an input line of the other matrix. Thus in these diagrams, one sums over all the possible labels of an edge that has no free ends. This corresponds directly to the formula for matrix multiplication where one sums the products of individual matrices over all double occurrences of indices.

Remark on the Three Strand Representation. Now look at the special case of a line graph with three nodes and two edges:

$$1 - - - - - 2 - - - - - 3.$$

The only admissible binary sequences are $|110\rangle$ and $|101\rangle$, so the space corresponding to this graph is two dimensional, and it is acted on by E_1 with $z(1) = 1$ in both cases (the empty walk terminates in the first node) and E_2 with $z(2) = 2$ for $|110\rangle$ and $z(2) = 2$ for $|101\rangle$. Then we have

$$E_1|110\rangle = 0, E_1|101\rangle = d|101\rangle,$$

$$E_2|xyz\rangle = |v\rangle\langle v|xyz\rangle$$

(xyz = 101 or 110) where $v = (\sqrt{1/d}, \sqrt{d - 1/d})^T$.

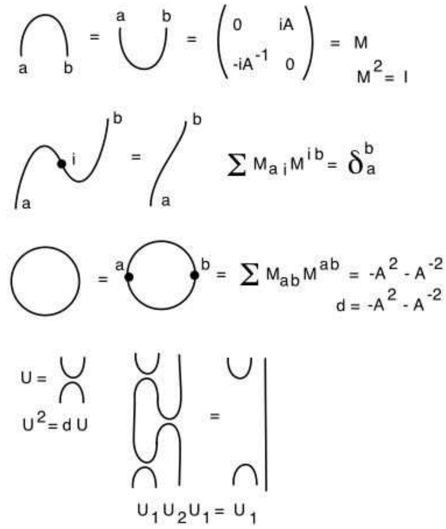


Figure 2. Diagrammatics for A Simple Temperley-Lieb Representation

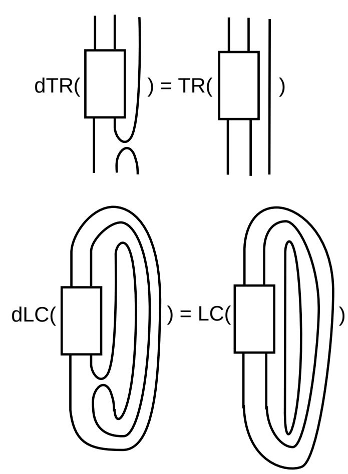


Figure 3. Trace Formula and Loop Count

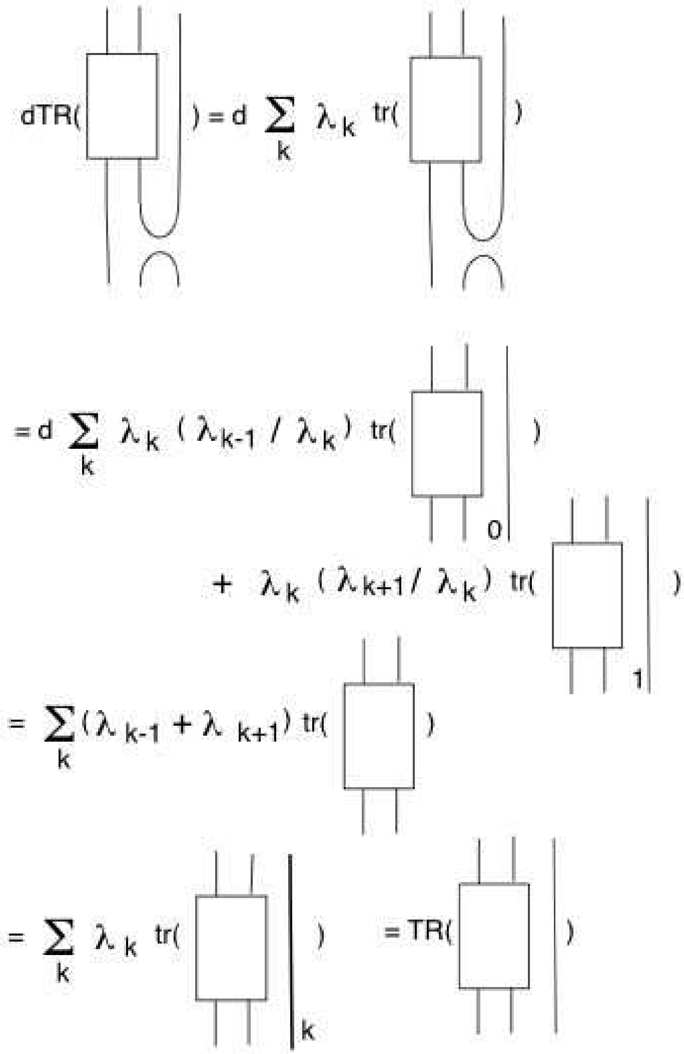


Figure 4. Proof of the Trace Formula

If one compares this two dimensional representation of the three strand Temperley - Lieb algebra and the corresponding braid group representation, with the representation Kauffman and Lomonaco use in their paper, it is clear that it is the same (up to the convenient replacement of $A = exp(i\theta)$ by $A = iexp(i\theta/2)$). The trace formula of AJL in this case is a variation of the trace formula that Kauffman and Lomonaco use. See the next section for a general discussion of the trace. Note that the AJL algorithm as formulated in[1] does not use the continuous range of angles that are available to the KL algorithm, but our generalization does allow this continuous angular range.

5. THE TRACE FUNCTION

We now treat the trace function on the AJL representation from a diagrammatic point of view. Let $\Phi : TL_n \longrightarrow Matr(H_{n,r})$ denote the representation of the Temperley- Lieb algebra discussed in the last section. Let $M = \Phi(\alpha)$ for any element α of the Temperley-Lieb algebra. We define a trace functional $TR(M)$ by the formula

$$Tr(M) = \sum_k \lambda_k tr(M_k)$$

where $\lambda_k = sin(k\theta)$ as in the previous section, tr denotes standard matrix trace and M_k denotes the restriction of M to walks that end on the node k in the graph G_r. Just as in AJL, since this trace can be used to compute the bracket polynomial of the closure of the braid at our admissible values of A on the unit circle, it follows that there is a quantum algorithm for this computation by having the quantum computer evaluate the standard traces $tr(M_k)$. Each M_k is a unitary matrix, and its trace can be found via the Hadamard test.

We will prove that

$$dTR(ME_n) = TR(M')$$

where M' is the inclusion of M in the corresponding representation of TL_{n+1} and E_n denotes the matrix representation of the element E_n in TL_{n+1}. This formula is exactly what is needed to have a Markov trace on the corresponding representation of the Artin braid group and hence to have a link invariant corresponding to the bracket model[7] of the Jones polynomial. To see this from the diagrammatic point of view, examine Figure 3. In that figure we have shown the diagrammatic version of the formula $dTR(ME_n) = TR(M')$ at the top of the figure, and we have compared with the corresponding loop count formula that corresponds to bracket polynomial calculation. The trace we define on the representation of the Temperley-Lieb algebra will agree with the loop counts performed by the bracket expansion on the closure of the braid, when the TR formula is satisfied. In the figure LC refers to loop-count and denotes the evaluation of a collection of loops as d^c where c is the number of loops in that collection.

Finally, Figure 4 is a diagrammatic rendering of the proof of the trace formula. The expansion in the middle of the derivation corresponds to the evaluation of a representation of a Temperley-Lieb projector as described in the last section, and it takes into account the action of this projector on the bitstrings in the representation. In the middle of the calculation the labels 0 and 1 remind the reader of the bitstring values for which these terms are non-zero. At the end of the calculation, the label k reminds the reader that these matrices are acting on walks that end on node k in the graph. We have left some of the details to the reader, but the main line of the argument is in Figure 4. The reader should note that the syntax of this use of the diagrams is explained in the last section. This completes the proof of the trace formula and hence the proof that this generalization of the AJL algorithm to continuous angular ranges can be used as a quantum algorithm for the Jones polynonmial.

REFERENCES

1. D. Aharonov, V. Jones, Z. Landau, A polynomial quantum algorithm for approximating the Jones polynomial, quant-ph/0511096.
2. D. Aharonov, I. Arad, The BQP-hardness of approximating the Jones polynomial, quant-ph/0605181
3. R.J. Baxter. Exactly Solved Models in Statistical Mechanics. Acad. Press (1982).
4. V.F.R. Jones, A polynomial invariant for links via von Neumann algebras, Bull. Amer. Math. Soc. **129** (1985), 103–112.
5. V.F.R.Jones. Hecke algebra representations of braid groups and link polynomials. Ann. of Math. 126 (1987), pp. 335-338.
6. V.F.R.Jones. On knot invariants related to some statistical mechanics models. Pacific J. Math., vol. 137, no. 2 (1989), pp. 311-334.
7. L.H. Kauffman, State models and the Jones polynomial, Topology **26** (1987), 395–407.
8. L.H. Kauffman, Statistical mechanics and the Jones polynomial, AMS Contemp. Math. Series **78** (1989), 263–297.
9. L.H. Kauffman and S. Lins, *Temperley-Lieb Recoupling Theory and Invariants of Three-Manifolds*, Princeton University Press, Annals Studies **114** (1994).
10. L.H. Kauffman, *Knots and Physics*, World Scientific Publishers (1991), Second Edition (1993), Third Edition (2002).
11. L.H. Kauffman, Quantum computing and the Jones polynomial, math.QA/0105255, in *Quantum Computation and Information*, S. Lomonaco, Jr. (ed.), AMS CONM/305, 2002, pp. 101–137.
12. L. H. Kauffman and S. J. Lomonaco Jr. q-deformed spin networks, knot polynomials and anyonic topological quantum computation. J. Knot Theory Ramifications 16 (2007), no. 3, 267–332.
13. L. H. Kauffman and S. J. Lomonaco Jr., The Fibonacci Model and the Temperley-Lieb Algebra. *International J. Modern Phys. B*, Vol. 22, No. 29 (2008), 5065-5080.
14. L. H. Kauffman and S. J. Lomonaco, Jr., A 3-Stranded Quantum Algorithm for the Jones Polynomial, Proc. SPIE, vol. 6573, (2007), 65730T-1-65730T-13. http://arxiv.org/abs/0706.0020
15. S. J. Lomonaco Jr. and L. H. Kauffman, A Quantum Manual for Computing the Jones Polynomial, Proc. SPIE on Quantum Information and Computation VI, Vol. 6976, (2008), pp. 69760K-1 to 69760K-4.
16. R. Marx, A. Fahmy, L. H. Kauffman, S. J. Lomonaco Jr., A Spörl, N. Pomplun, T. SchulteHerbrüggen, J. M. Meyers, and S. J. Glaser, Nuclear-magnetic-resonance quantum calculations of the Jones polynomial, *Physical Review A*, (2010).
17. R. Marx, A. Fahmy, L. H. Kauffman, S. J. Lomonaco Jr., A Spörl, N. Pomplun, T. SchulteHerbrüggen, J. M. Meyers, and S. J. Glaser, (In preparation.)
18. C. Nayak, E. H. Simon, A. Stern, M. Freedman, S. Das Sarma, Non-abelian anyons and topological quantum computation. Rev. Modern Phys. 80 (2008), no. 3, 1083–1159.
19. E. Witten. Quantum Field Theory and the Jones Polynomial. Comm. in Math. Phys. Vol. 121 (1989), 351-399.